Hate-Speech Classification of Code-Mixed Hinglish Text using Embedding-based BiLSTM and LSTM Models

Pankaj Biswas (222025043)
B.Tech CSE, 8th Semester, RSET, The Assam Royal Global University

Table of Contents

Title Page

Project Title: Hate-Speech Classification of Code-Mixed Hinglish Text using Embedding-based BiLSTM and LSTM Models

Individual Contribution to: *Developing a Sentiment Analysis Model for Code-Mixed Hindi-English (Hinglish) Text*

Student: Pankaj Biswas

Roll No.: 222025043

Course: B.Tech, Computer Science and Engineering, 8th Semester

Institution: Royal School of Engineering and Technology (RSET), The Assam Royal Global University, Guwahati

Guide: Dr. Dillip Rout, Assistant Professor, CSE

Session: 2022-2026

Abstract

This report presents the embedding-based recurrent-network track of a group project on binary hate-speech classification of code-mixed Hindi-English (Hinglish) text. Using the PRISM dataset (29,506 cleaned samples, non-hate vs hate), three pretrained word embeddings (GloVe, Word2Vec, FastText) were combined with a Bidirectional LSTM (BiLSTM) and a many-to-one LSTM, and evaluated under regular, language-wise, and multi-stage language-training regimes. Models were assessed with Accuracy, Balanced Accuracy, Precision, Recall, Specificity, F1, and AUC-ROC on a held-out test set. The best configuration, GloVe+BiLSTM trained with a multi-stage language curriculum and evaluated on the combined set, achieved an **F1 of 0.8041 and an AUC-ROC of 0.9139** (Accuracy 0.8204), outperforming the Word2Vec and FastText variants and the transformer baselines reported in the same evaluation. The results show that, for low-

resource code-mixed text, a multi-stage training curriculum is the single largest driver of performance for hybrid embedding-BiLSTM models.

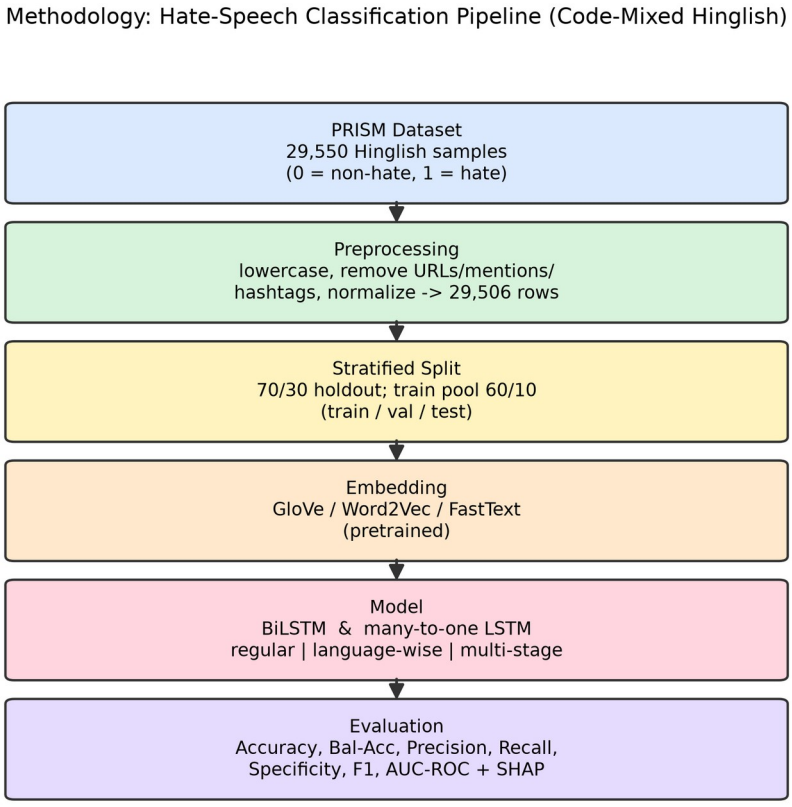
1. Introduction

Code-mixed languages such as Hinglish, an informal blend of Hindi and English written largely in Latin script, are pervasive on social media yet difficult to model: grammar is inconsistent, transliteration varies, and annotated resources are scarce. Reliable hate-speech detection on such text is socially important for content moderation but is under-served by models built for monolingual, well-formed input.

This contribution studies how the **choice of pretrained embedding** and the **training curriculum** affect hate-speech classification on the PRISM dataset, holding the recurrent architecture fixed. The dataset contains 29,550 raw (29,506 cleaned) Hinglish samples labelled non-hate (0) or hate (1), sourced from Kaggle. Three embeddings (GloVe, Word2Vec, FastText) are paired with a BiLSTM and a many-to-one LSTM, and trained under three regimes, to identify the configuration most robust to the linguistic noise of code-mixed text.

2. Methods

2.1 Pipeline



Methodology pipeline

2.2 Preprocessing and data split

The raw corpus was cleaned by lowercasing; removing URLs, mentions, and hashtags; normalizing elongated words and whitespace; and removing duplicate rows, reducing 29,550 to 29,506 samples. Retained features were clean_text, hate_label, language, text_length, and word_count. The data was divided with a **stratified 70/30 holdout**; the 70 percent training pool was further split **60/10** into training and validation. Language-wise class balance was preserved across all subsets to reduce sampling bias.

Category	Combined	English	Hindi	Hinglish
Total samples	29,506	14,994	9,738	4,774
Non-hate (0)	15,799	7,495	5,393	2,911
Hate (1)	13,707	7,499	4,345	1,863

2.3 Features

Inputs are tokenized clean_text sequences mapped to a pretrained embedding matrix. Three embeddings were compared: **GloVe**, **Word2Vec**, and **FastText** (subword), the last expected to better handle transliteration variants of Hinglish.

2.4 Models

- **BiLSTM** over pretrained embeddings: embedding_dim 300, hidden_dim 256, dropout 0.5, max_seq_len 128.
- **Many-to-one LSTM**: embedding_dim 100, hidden_dim 128, dropout 0.2, max_seq_len 100.

Three training regimes were evaluated: **regular** (single combined fit), **language-wise regular** (per-language strategies: English, Hindi, Hinglish, combined), and **multi-stage language training** (sequential fine-tuning across languages, e.g. English to Hinglish to Hindi to Full), including a six-variation ordering sweep on GloVe+BiLSTM.

2.5 Hyperparameter selection

Hyperparameters were set per architecture (Section 3) rather than via automated grid search; the values follow the configurations validated for the group project. The multi-stage ordering sweep acts as a structured search over training curricula.

3. Experiment Setup

- **Hardware/Software:** Google Colab (NVIDIA T4 GPU), Python 3.10, TensorFlow/Keras 2.15, scikit-learn, Gensim (embeddings), SHAP (explainability), Matplotlib.
- **Data split:** stratified 70/30 holdout; train pool split 60/10 into train/validation; fixed random seed for reproducibility; language-wise balance preserved.
- **Cross-validation:** a fixed stratified hold-out split was used rather than k-fold CV.

- **Training settings (BiLSTM):** Adam, lr 1e-3, weight_decay 0, 10 epochs, batch size 32, binary cross-entropy.
- **Training settings (LSTM):** Adam, lr 1e-2, weight_decay 0, 30 epochs, batch size 16.
- **Reproducibility:** notebooks, configs (YAML), trained models (.h5), and result tables (CSV) are provided in code/, configs/, models/, and tables/.

4. Results

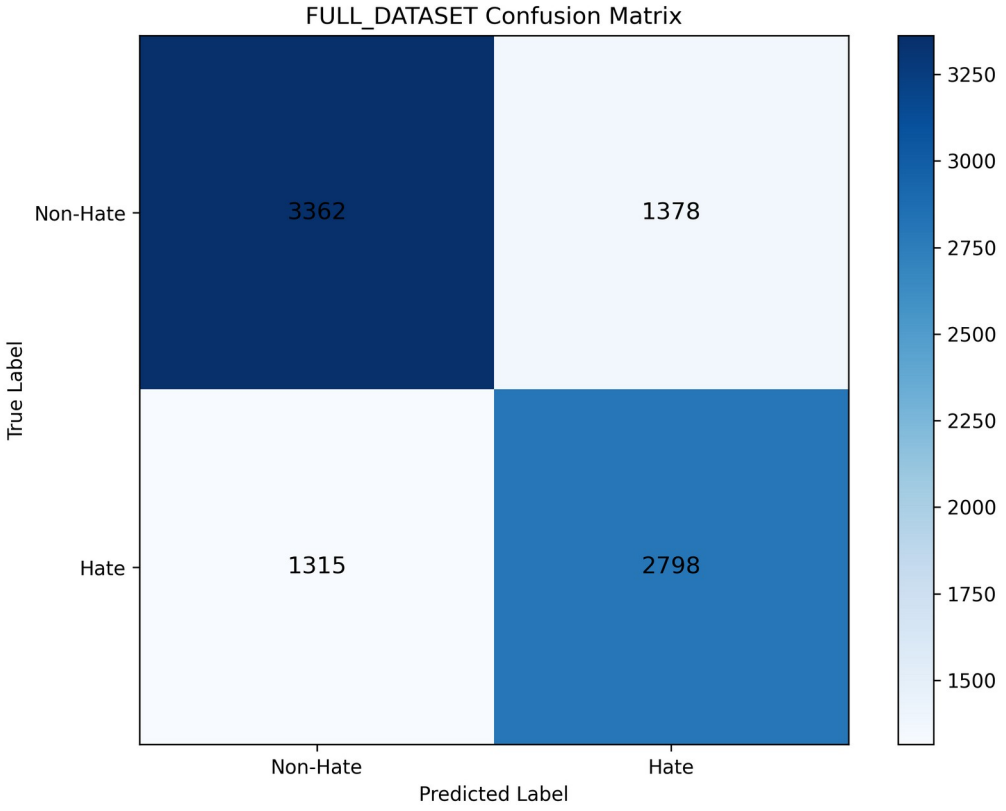
4.1 Best model

Model	Regime	Strategy	Accuracy	F1	AUC-ROC
GloVe+BiLSTM	multi-stage	combined	0.8204	0.8041	0.9139

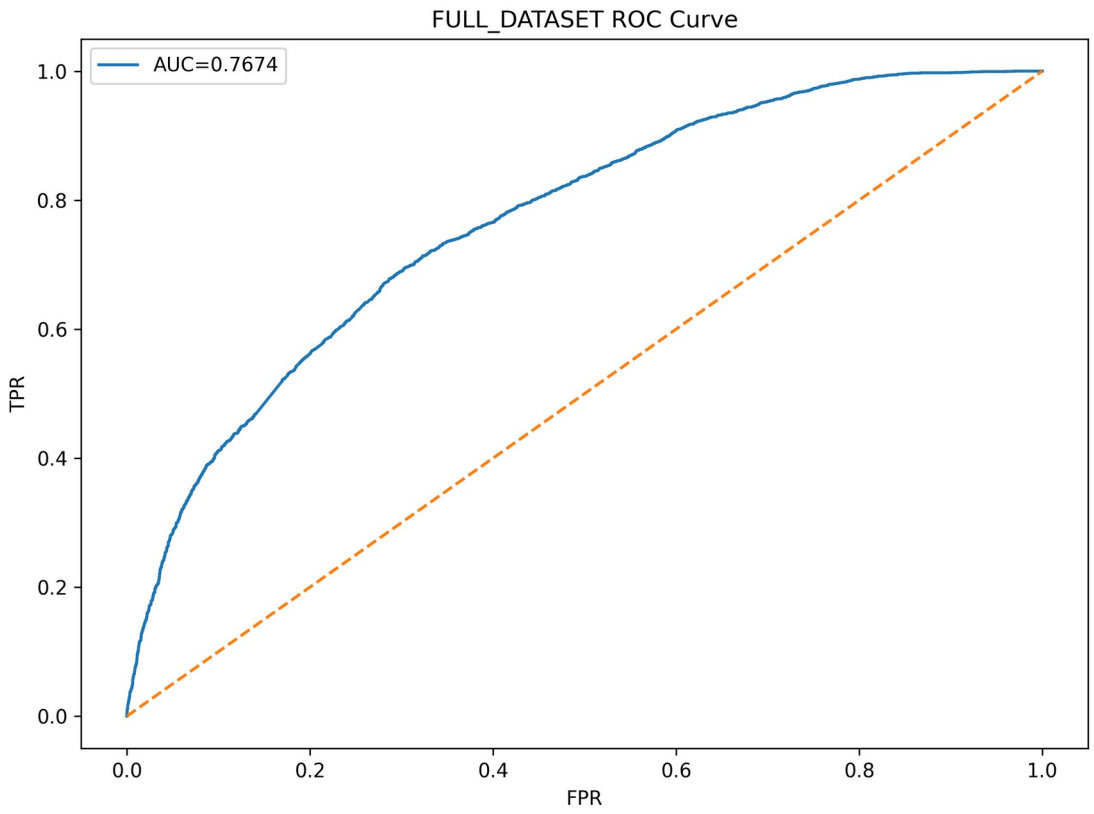
4.2 Model comparison (test set, combined strategy)

Model	Regime	Accuracy	Bal-Acc	Precision	Recall	Specificity	F1	AUC-ROC
GloVe+BiLSTM	multi-stage	0.8204	0.8186	0.8149	0.7935	0.8437	0.8041	0.9139
Word2Vec+BiLSTM	multi-stage	0.7305	0.7267	0.7264	0.6734	0.7800	0.6989	0.8091
FastText+BiLSTM	multi-stage	0.7018	0.7007	0.6767	0.6856	0.7158	0.6811	0.7705
GloVe+LSTM	regular	0.6779	0.6627	0.7591	0.4491	0.8763	0.5644	0.7532
Word2Vec+LSTM	regular	0.6675	0.6573	0.6914	0.5133	0.8012	0.5892	0.7294
FastText+LSTM	regular	0.6610	0.6496	0.6906	0.4895	0.8097	0.5729	0.7208

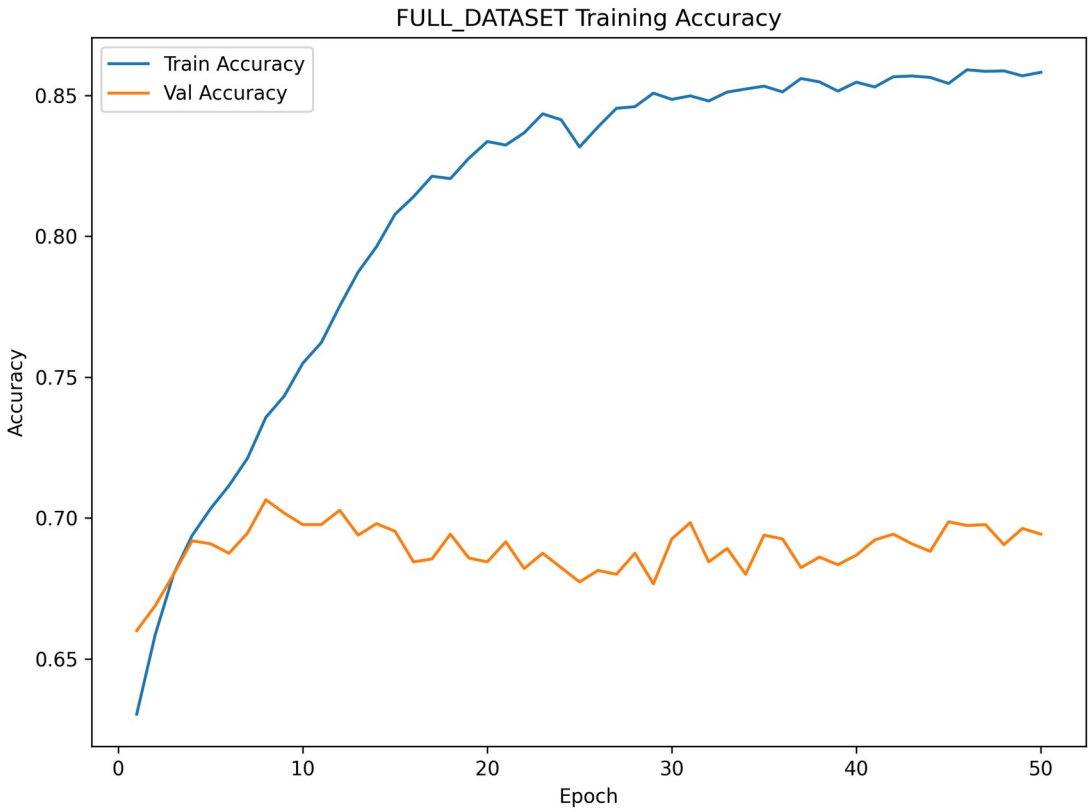
4.3 Performance plots (GloVe+BiLSTM)



Confusion matrix



ROC curve



Training and validation accuracy

4.4 Explainability (SHAP)

SHAP analysis on GloVe+BiLSTM shows the model attends to profanity and abuse tokens for the hate class. False negatives concentrate on obfuscated or romanized Hindi abuse and code-switch boundaries; false positives on aggressive but non-hateful phrasing.

5. Discussion

Findings. The multi-stage language curriculum is the dominant factor: GloVe+BiLSTM improves from roughly 0.64 F1 under language-wise training to 0.804 F1 (0.914 AUC) under multi-stage combined, surpassing the Word2Vec and FastText variants. Starting the curriculum from the largest, cleanest subset (English) and ending on the full combined set produced the strongest ordering.

Errors and limitations. GloVe+BiLSTM collapses on the Hindi-only strategy (it predicts all non-hate: precision, recall, and F1 of 0), a minority-language failure mode rather than a working result. The many-to-one LSTM variants show high specificity but low recall, leaning toward the majority non-hate class. A fixed hold-out split was used rather than k-fold cross-validation.

Bias. Class balance is preserved across splits, but the language imbalance (English 50.8 percent, Hindi 33.0 percent, Hinglish 16.2 percent) biases combined models toward English patterns, which the language-wise and multi-stage strategies partly mitigate.

Future work. Add k-fold cross-validation and PR-AUC reporting; improve preprocessing for romanized/obfuscated Hindi abuse; address the Hindi-only collapse with class re-weighting or focal loss; and compare against contextual transformer encoders.

Code and Data Availability

All code, trained models, configurations, result tables, figures, and data used in this work are publicly available on Hugging Face.

- **Model repository (all code, models, figures, tables, configs, reports):** https://huggingface.co/pankajbiswas6/hinglish-hate-speech-bilstm
- **Dataset repository (PRISM cleaned data + train/validation/test splits):** https://huggingface.co/datasets/pankajbiswas6/prism-hinglish-hate-speech

Notebooks (code) by phase:

- Phase 1 (dataset split, FastText): https://huggingface.co/pankajbiswas6/hinglish-hate-speech-bilstm/tree/main/phase1/notebooks
- Phase 2 (hybrid baselines, monolingual): https://huggingface.co/pankajbiswas6/hinglish-hate-speech-bilstm/tree/main/phase2/notebooks
- Phase 4 (many-to-one LSTM): https://huggingface.co/pankajbiswas6/hinglish-hate-speech-bilstm/tree/main/phase4/notebooks
- Phase 6 (regular / sequential / multi-stage, six variations): https://huggingface.co/pankajbiswas6/hinglish-hate-speech-bilstm/tree/main/phase6/notebooks

Trained models (.h5):

- Phase 3: https://huggingface.co/pankajbiswas6/hinglish-hate-speech-bilstm/tree/main/phase3/models
- Phase 4: https://huggingface.co/pankajbiswas6/hinglish-hate-speech-bilstm/tree/main/phase4/models
- Phase 5: https://huggingface.co/pankajbiswas6/hinglish-hate-speech-bilstm/tree/main/phase5/models
- Phase 6: https://huggingface.co/pankajbiswas6/hinglish-hate-speech-bilstm/tree/main/phase6/models

Other resources:

- Configurations (YAML): https://huggingface.co/pankajbiswas6/hinglish-hate-speech-bilstm/tree/main/configs
- Result tables (CSV): https://huggingface.co/pankajbiswas6/hinglish-hate-speech-bilstm/tree/main/tables

- Figures (PNG): https://huggingface.co/pankajbiswas6/hinglish-hate-speech-bilstm/tree/main/figures
- SHAP explainability plots: https://huggingface.co/pankajbiswas6/hinglish-hate-speech-bilstm/tree/main/phase5/shap

References

1. Gaurav Singh. Sentiment Analysis of Code-Mixed Social Media Text (Hinglish). arXiv, 2021.
2. Varsha Thakur, Roshani Sahu, Somya Omer. Current State of Hinglish. SSRN Electronic Journal, 2020.
3. Neha Agarwal et al. Improving Sentiment Analysis. Educational Administration: Theory and Practice, 2024.
4. Gadde Satya Sai Naga Himabindu et al. A self-Attention hybrid model for code-mixed language. Social Network Analysis and Mining, 2022.
5. Pennington, Socher, Manning. GloVe: Global Vectors for Word Representation. EMNLP, 2014.
6. Mikolov et al. Efficient Estimation of Word Representations in Vector Space (Word2Vec). 2013.
7. Bojanowski et al. Enriching Word Vectors with Subword Information (FastText). TACL, 2017.
8. Lundberg, Lee. A Unified Approach to Interpreting Model Predictions (SHAP). NeurIPS, 2017.